

REMARKS

Applicant respectfully requests reconsideration and allowance of the present application. Claims 1-25, 27-31, 34-35, and 37-43 are pending in this application.

Interview Request

Applicant requests an Interview with the Examiner prior to issuing a response to this Amendment.

35 USC § 101

Claims 1-9, 11-18, 20-26 and 40-43 stand rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Independent claims 1, 11, 20 and 40 have been amended to state that the method is implemented by a computer. Accordingly, Applicant submits that claims 1-9, 11-18, 20-26 and 40-34 satisfy 35 U.S.C. 101.

Applicant respectfully requests that the §101 rejection be withdrawn.

Claim Objections

Claim 16 is objected to as incorrectly referring to the wrong base claim. Applicant has amended claim 16 to properly refer to claim 11. Applicant submits that claim 16 is now in proper form.

35 USC § 102

Claims 1-8, 10-19 and 28-43 stand rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,336,139 to Feridun et al. (hereinafter "Feridun").

As amended, Claim 1 recites:

A computer-implemented method comprising:
receiving a plurality of events;
applying the plurality of events to a correlation function, wherein the correlation function is implemented as a state machine and is configured to correlate the plurality of events;
identifying an event to which an update consumer has subscribed, wherein the update consumer is associated with the state machine;
applying the update consumer to the state machine in response to the identified event; and
generating a specific event if the correlation function is satisfied by the plurality of events.

Thus, an update consumer is applied to a state machine when a particular event occurs. Although the Feridun reference discloses a method for event correlation, Feridun fails to disclose the use of an update consumer and a state machine in the manner recited in amended claim 1. Feridun does mention the use of state machines, but fails to disclose the updating of state machines using an update consumer, as recited in claim 1. Accordingly, Feridun fails to disclose the elements of claim 1.

For at least the reasons stated above, Applicant respectfully submits that claim 1 is not anticipated by Feridun and is allowable. Given that claims 2-8 and 10 depend from claim 1, those claims are also allowable for at least the same reasons.

As amended, Claim 11 recites:

A computer-implemented method comprising:
receiving a plurality of events;
receiving a plurality of data elements;
identifying a plurality of correlation functions configured to correlate the plurality of events and the plurality of data elements, wherein each correlation function is implemented with an associated state machine, and wherein each state machine has an associated update consumer that updates the state of the associated state machine;
applying the plurality of events and the plurality of data elements to the plurality of correlation functions; and
generating a specific event if at least one of the plurality of correlation functions is satisfied.

As discussed above with respect to claim 1, Feridun fails to disclose a state machine that is updated by an associated update consumer. In particular, Feridun fails to disclose "wherein each correlation function is implemented with an associated state machine, and wherein each state machine has an associated update consumer that updates the state of the associated state machine;", as recited in amended claim 11. Applicant submits that Feridun does not disclose the elements of claim 11.

For at least the reasons stated above, Applicant respectfully submits that claim 11 is not anticipated by Feridun and is allowable. Given that claims 12-19 depend from claim 11, those claims are also allowable for at least the same reasons.

As amended, Claim 28 recites:

An apparatus comprising:
a plurality of event consumers; and
an event correlator coupled to the plurality of event consumers, the event correlator to receive events from at least one event source and to receive data elements from at least one data source, the event correlator further to receive at least one correlation function configured to correlate events and data elements and to apply the received events and the received data elements to the correlation function, wherein the correlation function is implemented by a state machine having an associated update consumer that updates the state of the state machine, and wherein the event correlator generates a specific event if the received events and the received data satisfy the correlation function.

As discussed above, Feridun does not disclose a state machine that is updated by an associated update consumer. In particular, Feridun fails to disclose “wherein the correlation function is implemented by a state machine having an associated update consumer that updates the state of the state machine”, as recited in amended claim 28. Accordingly, Applicant submits that Feridun does not disclose the elements of claim 28.

Thus, Applicant respectfully submits that claim 28 is not anticipated by Feridun and is allowable for at least the reasons stated above. Given that claims 29-31 and 34 depend from claim 28, those claims are also allowable for at least the same reasons.

As amended, Claim 35 recites:

One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:

- receive a plurality of events;
- identify a plurality of correlation functions configured to correlate the plurality of events, wherein each of the plurality of correlation functions is implemented as a state machine having an associated update consumer;
- apply the plurality of events to the plurality of correlation functions to determine whether any of the plurality of correlation functions are satisfied by the plurality of events, wherein the plurality of events are applied by causing update consumers associated with each event to update the state of the associated state machine; and
- generate a specific event if one of the plurality of correlation functions is satisfied by the plurality of events.

As discussed above, the Feridun reference does not disclose a state machine that is updated by an associated update consumer. In particular, Feridun fails to disclose “wherein each of the plurality of correlation functions is implemented as a state machine having an associated update consumer;”, as recited in amended claim 35. Accordingly, Applicant submits that Feridun does not disclose the elements of claim 35.

Thus, Applicant respectfully submits that claim 35 is not anticipated by Feridun and is allowable for at least the reasons stated above. Given that claims 37-39 depend from claim 35, those claims are also allowable for at least the same reasons.

As amended, Claim 40 recites:

A computer-implemented method comprising:
receiving events from event providers;
creating a first state machine;
creating a second state machine;
associating a first event type with the first state machine, wherein the first state machine has an associated first update consumer to update the state of the first state machine;
associating a second event type with the second state machine, wherein the second state machine has an associated second update consumer to update the state of the second state machine;
in response to receiving an event having a first event type, applying the first update consumer to the first state machine;
in response to receiving an event having a second event type, applying the second update consumer to the second state machine; and
if the events are correlated:
generating an additional event; and
sending the additional event to an event consumer.

As discussed above, Feridun does not disclose a state machine that is updated by an associated update consumer. Further, Feridun does not disclose creating first and second state machines, each of which is associated with a particular event type and further associated with a particular update consumer to update the state of the state machine. Accordingly, Applicant submits that Feridun fails to disclose the elements of claim 40.

Thus, Applicant respectfully submits that claim 40 is not anticipated by Feridun and is allowable for at least the reasons stated above. Given that claims 41-43 depend from claim 40, those claims are also allowable for at least the same reasons.

35 USC § 103

Claims 9 and 20-27 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Feridun in view of “Java Developer’s Guide” by Jaworski (hereinafter “Jaworski”).

As amended, claim 9 recites, “A method as recited in claim 8 further including deleting the instance of the particular state machine if the instance of the particular state machine reaches a final state.” Applicant submits that Feridun fails to disclose the elements of claim 1 (from which claim 9 depends) as discussed above. Further, Applicant submits that Feridun fails to suggest the elements of claim 1. Although Feridun discusses the use of state machines, Feridun makes no mention of using a update consumer with the state machine. As such, Feridun provides no suggestion to use an update consumer as recited in claim 1.

Further, Feridun fails to disclose or suggest “deleting the instance of the particular state machine if the instance of the particular state machine reaches a final state”, as recited in claim 9. Although Feridun mentions stat machines, the reference fails to disclose the deletion of an instance of a particular state machine as claimed in claim 9.

The Jaworski reference fails to correct the deficiencies of the Feridun reference with respect to claim 9. Chapter 3 of Jaworski discusses use of the Java Developer’s Kit. However, the reference fails to disclose or suggest the use of state machines, update consumers and deleting of state machine instances as contained in claim 9 (as well as base claim 1 and intervening claim 8).

Thus, Applicant respectfully submits that claim 9 is allowable over Feridun in view of Jaworski for at least the reasons stated above.

As amended, Claim 20 recites:

A computer-implemented method comprising:
identifying a schema for creating state machines, the state machines to correlate at least two events;
creating an instance of a particular state machine;
defining transitions for the particular state machine by subscribing to at least one event; and
applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object.

The Feridun reference fails to disclose or suggest “applying an update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object”, as recited in amended claim 20. As discussed above, Feridun does not disclose a state machine that is updated by an associated update consumer. Further, Feridun does not suggest this type of operation. Additionally, Feridun does not disclose or suggest the use of an update consumer that is a class object.

The Jaworski reference fails to correct the deficiencies of the Feridun reference with respect to claim 20. Although Chapter 3 of the Jaworski reference discloses a Java Developer’s Kit, Jaworski fails to disclose or suggest an update consumer that updates the state of a state machine, where the update consumer is a class object, as recited in claim 20.

Thus, Applicant respectfully submits that claim 20 is allowable over Feridun in view of Jaworski for at least the reasons stated above. Given that

claims 21-25 and 27 depend from claim 20, those claims are also allowable for at least the same reasons.

Conclusion

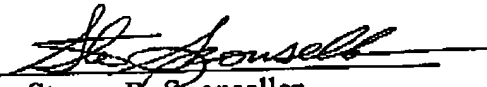
Claims 1-25, 27-31, 34-35, and 37-43 are in condition for allowance.

Applicant respectfully requests the issuance of the subject application.

Respectfully Submitted,

Date: 10-06-05

By:


Steven R. Sponseller
Reg. No. 39,384
(509) 324- 9256 ext. 250